REMARKS

In response to the Office Action mailed December 21, 2010, reconsideration is respectfully requested. To further the prosecution of this application, amendments have been made in the claims, and each of the rejections set forth in the Office Action has been carefully considered and is addressed below. The claims as presented are believed to be in condition for allowance.

Claims 1-21 were previously pending in this application. Claims 1, 8 and 15 are amended. No claims are added or cancelled. As a result, claims 1-21 remain pending for examination, with claims 1, 8 and 15 being independent. No new matter has been added.

Claim Rejections Under 35 U.S.C. §103

Each of independent claims 1, 8 and 15 is rejected under 35 U.S.C. §103(a) as purportedly being obvious over U.S. Patent No. 5,692,792 to Sudo in view of an article entitled "TASK Forces: Distributed Software for Solving Problems of Substantial Size," Proceedings of 4th Software Engineering, September, 1979 to Jones et al. ("Jones"). As presented herein, each of independent claims 1, 8 and 15 patentably distinguishes over the prior art of record.

A. Brief Overview Of Embodiments Of The Invention

Computer-based operations that involve complex interactions between different computers, such as those supporting various networking protocols and distributed transactions, often have to be extensively tested to ensure that they function properly (see Applicant's specification at, e.g., ¶[0002]). Typically, to test a given type of operation, a set of test computers are chosen to run a test case, which includes a sequence of tasks to be performed by different test computers in an interactive fashion (¶[0002]).

In conventional systems, execution of a test run is controlled by a centralized controller, which proceeds through the sequence of tasks one by one, so that a first test computer is instructed to carry out one task, the controller receives the result of that processing, and then instructs another test computer to carry out the next task in the sequence based on the outcome of the previous task

(¶[0002]). The use of a centralized controller to control automated execution of distributed computer tasks places a significant constraint on the availability of test computers, as each test computer is required to form a communication link with the centralized test controller to receive instructions and report task execution status end results (¶[0003]).

Some embodiments of the invention provide a framework for automated dynamic execution of distributed tasks without a centralized controller to coordinate the execution of the tasks (¶[0004]). For example, in some embodiments, execution of a sequence of tasks is coordinated through the cooperation of peer computers selected to perform the tasks (¶[0004]). In some embodiments, each peer computer includes an "execution agent" for handling the execution of its assigned tasks (¶[0004]). A first computer in the group of peer computers may receive execution instructions, which it then passes on to other peer computers so that each computer knows which tasks to perform and their tasks' relationships to those performed by other computers (¶[0004]). The execution agents of the peer computers communicate during execution to synchronize the execution of the tasks and determine execution status (¶[0004]).

The foregoing summary is provided to assist the Examiner in appreciating various aspects of the present invention. However, this summary does not necessarily apply to each independent claim, and the language of each independent claim may differ in material respects from the summary above. Thus, Applicant respectfully requests that careful consideration be given to the language of each independent claim, and they each be addressed on its own merits, without relying on the summary above. In this respect, Applicant does not rely on the foregoing summary to distinguish any of the claims over the prior art, but rather relies only upon the language of the claims and the arguments presented below.

B. Rejection of independent claim 1

Amended claim 1 recites a computer-readable medium having computer-executable instructions stored thereon which, when executed by a first computer in a group of computers, performs steps for coordinated execution of distributed tasks. The steps comprise, *inter alia*: receiving from at least one server, by the first computer in the group of peer computers, a set of execution instructions for the peer computers including a sequence of tasks to be performed and an automatically, without user action, determined assignment of the tasks to the peer computers; forwarding, by the first computer to the other peer computers in the group, execution instruction information, derived from the execution instructions, informing each respective peer computer of a dependency between one or more tasks assigned to the respective peer computer and one or more tasks assigned to the other peer computers in the group of peer computers; executing, by the first computer, a portion of the sequence of tasks; and receiving, by the first computer from each of the other peer computers, and transmitting, by the first computer to each of the other peer computers, peer-to-peer communication messages containing task execution status information to synchronize and coordinate the execution of the sequence of tasks, the task execution status information comprising an indication that one task upon which another task depends has completed.

Support for the amendments to claim 1 may be found in the specification at, for example, ¶¶[0022] and [0029].

Amended claim 1 patentably distinguishes over any combination of the asserted references, as neither Sudo nor Jones discloses or suggests a first computer forwarding execution instruction information informing a peer computer of a dependency between one or more tasks assigned to the peer computer and one or more tasks assigned to other peer computers, as recited by claim 1. In addition, neither Sudo nor Jones discloses or suggests the first computer receiving and transmitting peer-to-peer communication messages containing task execution status information that comprises an indication that one task upon which another task depends has completed, as recited by claim 1.

Sudo, which the Office Action contends satisfies many of the limitations of formerly pending claim 1, discloses a technique for distributing the load a task imposes on a heavily loaded node by "expanding" the task so that it is also executed on a lightly loaded node (Abstract). Sudo teaches that "expanding" a task involves transferring process threads to a lightly loaded node (col. 5, lines 20-22).

However, Sudo says nothing about dependencies between tasks or process threads. In this respect, the disclosure of Sudo is limited to monitoring the load imposed on various nodes by a particular processing task, and distributing that load when appropriate (see, e.g., col. 7, lines 7-10).

Sudo says nothing about any sort of task-interdependency, and so Sudo necessarily does not teach or suggest a first computer forwarding execution instruction information informing a peer computer of a dependency between one or more tasks assigned to the peer computer and one or more tasks assigned to another peer computer, as recited by claim 1. Sudo also does not teach or suggest a first computer receiving and transmitting peer-to-peer communication messages containing task execution status information which comprises an indication that one task upon which another task depends has completed, as recited by claim 1.

Furthermore, one skilled in the art would not have modified Sudo's system to execute tasks between which there are such dependencies. Sudo teaches distributing the load imposed by a single task on a node by breaking that task down into its constituent threads, and transferring the threads to one or more other nodes (col. 5, lines 20-22). Sudo makes no allowance for any dependencies between threads, or between overall tasks. To modify Sudo to allow for such interdependencies would fundamentally alter the manner in which the Sudo system operates, because doing so would place restrictions on which process threads could be transferred to another node, and when transfer could occur. For example, one thread that is dependent on another could only be transferred to another node for processing after the other thread is completed. Enabling the Sudo system to handle task and thread interdependencies would entail a fundamental redesign. MPEP §2143.01(VI) is clear that if a proposed modification would change the principle of operation of the reference being modified, and such a change to the Sudo system clearly would, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. MPEP §2143.01(VI).

In view of the foregoing, claim 1 patentably distinguishes over the prior art of record. Accordingly, the rejection of claim 1 under 35 U.S.C. §103(a) as purportedly being obvious over Sudo in view of Jones should be withdrawn.

Claims 2-7 depend from claim 1 and are allowable for at least the same reasons.

C. Rejection Of Independent Claim 8

Claim 8 recites a method of performing coordinated execution of distributed tasks by a group of peer computers, comprising: receiving from at least one server, by a first computer in the group of peer computers, each of the peer computers in the group of peer computers having at least one processing unit and one input device and one output device distinct from the at least one processing unit, a set of execution instructions for the peer computers, the execution instructions including a sequence of tasks to be performed and an automatically, without user action, determined assignment of the tasks to the peer computers; forwarding, by the first computer in the group of peer computers to the other peer computers in the group, execution instruction information derived from the execution instructions, the execution instruction information that is forwarded to each respective peer computer in the group informing the respective peer computer of a dependency between one or more tasks assigned to the respective peer computer and one or more tasks assigned to the other peer computers in the group of peer computers; executing, by the first computer, a portion of the sequence of tasks assigned thereto in connection with execution of other of the sequence of tasks assigned to the other peer computers in the group; and receiving, by the first computer from each of the other peer computers, and transmitting, by the first computer to each of the other peer computers, peer-to-peer communication messages containing task execution status information to synchronize and coordinate the execution of the sequence of tasks, the task execution status information comprising an indication that one task upon which another task depends has completed.

It should be appreciated from the discussion above regarding claim 1 that Sudo does not disclose or suggest a first computer forwarding execution instruction information informing a respective peer computer of a dependency between one or more tasks assigned to the peer computer and one or more tasks assigned to other peer computers in a group of peer computers, or the first computer receiving and transmitting peer-to-peer communication messages containing task execution status information comprising an indication that one task upon which another task depends has completed, as recited by claim 8. It should further be appreciated that modifying the Sudo system to allow for task interdependencies would require changing the system's principle of operation, so that the teachings of the references are not sufficient to render claim 8 *prima facie* obvious.

Accordingly, claim 8 patentably distinguishes over the prior art of record, such that the rejection of claim 8 under 35 U.S.C. §103(a) as purportedly being obvious over Sudo in view of Jones should be withdrawn.

Claims 9-14 depend from claim 8 and are allowable for at least the same reasons.

D. Rejection Of Independent Claim 15

Claim 15 recites a computer system for performing automated execution of distributed tasks, comprising: a plurality of peer computers connected by a network, each of the plurality of peer computers having: at least one processing unit; at least one input device distinct from the at least one processing unit; at least one output device distinct from the at least one processing unit; and an execution agent, the execution agent of each peer computer being programmed for: receiving a set of execution instructions for the peer computers, the execution instructions including a sequence of tasks to be performed and an automatically, without user action, determined assignment of the tasks to the peer computers; forwarding to the execution agents on the other peer computers execution instruction information derived from the execution instructions, the execution instruction information forwarded to each respective peer computer informing the respective peer computer of a dependency between one or more tasks assigned to the respective peer computer and one or more tasks assigned to another of the plurality of peer computers; executing a portion of the sequence of tasks in connection with execution of other of the sequence of tasks assigned to the other peer computers; and receiving from each of the other peer computers and transmitting to each of the other peer computers peer-to-peer communication messages containing task execution status information to synchronize and coordinate the execution of the sequence of tasks, the task execution status information comprising an indication that one task upon which another task depends has completed.

It should be appreciated from the discussion above regarding claim 1 that Sudo does not disclose or suggest an execution agent programmed for forwarding instruction information informing a respective peer computer of a dependency between one or more tasks assigned to the peer computer and one or more tasks assigned to another of a plurality of computers; or receiving from and transmitting to other peer computers peer-to-peer communication messages containing task execution status information comprising an indication that one task upon which another task depends has completed, as recited by claim 15. It should further be appreciated that modifying the Sudo system to allow for task interdependencies would require changing the system's principle of operation, so that the teachings of the references are not sufficient to render claim 15 *prima facie* obvious.

Accordingly, claim 15 patentably distinguishes over the prior art of record, such that the rejection of claim 15 under 35 U.S.C. §103(a) as purportedly being obvious over Sudo in view of Jones should be withdrawn.

Claims 16-21 depend from claim 15 and are allowable for at least the same reasons.

CONCLUSION

A Notice of Allowance is respectfully requested. The Examiner is requested to call the undersigned at the telephone number listed below if this communication does not place the case in condition for allowance.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, the Director is hereby authorized to charge any deficiency or credit any overpayment in the fees filed, asserted to be filed or which should have been filed herewith to our Deposit Account No. 23/2825, under Docket No. M1103.70583US00.

Dated: 2-23-11

Respectfully submitted,

By: Randy J. Pritzker

Randy J. Pritzker
Registration No.: 35,986
WOLF, GREENFIELD & SACKS, P.C.
600 Atlantic Avenue
Boston, Massachusetts 02210-2206
617.646.8000